Formerly by Motorola
freescale semiconductor

FACSIMILE

LAW DEPARTMENT
FREESCALE SEMICONDUCTOR, INC.

DATE: June 24, 2005

TO:	MS: Appeal Briefs - Patents	
	(ADDRESSEE)	(EXTENSION)
	USPTO	(703) 872-9306
	(LOCATION)	(FAX NUMBER)
FROM:	Stacie Herrera for James L. Clingan, Jr.	(512) 996-6848
	(SENDER)	(EXTENSION)

TOTAL NUMBER OF PAGES 16 **(including this page)**

IF YOU HAVE ANY TROUBLE OR QUESTIONS WITH TRANSMISSION, OR HAVE RECEIVED IT IN ERROR, PLEASE CALL:(512) 996-6839

ALL ITEMS MARKED WITH AN "X" ARE INCLUDED:

1.	x	1 page Facsimile Cover Sheet
2.	x	15 page Appeal Brief

I HEREBY CERTIFY THAT THIS CORRESPONDENCE IS BEING FACSIMILE TRANSMITTED TO THE PATENT AND TRADEMARK OFFICE:

ON: 6/24/05 (Date) Stacie Herrera (Signature)

FREESCALE LAW DEPARTMENT
7700 W. PARMER LANE MD: TX32/PL02
AUSTIN, TEXAS 78729
Fax Number (512) 996-6854

PLEASE GIVE THESE PAPERS TO:

EXAMINER: Matthew T. Henning
GROUP ART UNIT: 2131
SERIAL NO.: 09/725,821
FILED: NOVEMBER 29, 2002
INVENTOR: JAMES D. DWORKIN ET AL

In re Application of:
James D. Dworkin, et al.
Serial No.: 09/725,821
Filed: November 29, 2002
For: A CIRCUIT FOR GENERATING HASH VALUES

June 24, 2005

Art Unit: 2131
Examiner: Matthew T. Henning
Docket No.: SC11015ZP

Certificate of Transmission under 37 CFR 1.8

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office.

on June 24, 2005

Signature

Stacie Herrera
Printed Name of Person Signing Certificate

APPEAL BRIEF

COMMISSIONER FOR PATENTS
ALEXANDRIA, VA 22313
BOARD OF PATENT APPEALS & INTERFERENCES:

This brief is filed in the matter of the Appeal to the Board of Appeals and Interferences of the rejection of the claims of the above-referenced application for patent.

REAL PARTY IN INTEREST

The present application is wholly assigned to FREESCALE SEMICONDUCTOR, INC., with its headquarters in Austin Texas.

RELATED APPEALS AND INTERFERENCES

Appellants are unaware of other appeals or interferences which will directly affect, be directly affected by, or have a bearing on the Board's decision in this appeal.

STATUS OF CLAIMS

Claims 1-8 and 14-18 are pending. Claims 9-13 have been canceled.

Claims 1-7, 14, 15, 17, and 18 stand rejected under 35 U.S.C. 103(a) as being obvious in view of Ober et al., U.S. Patent No. 6,708,273 (Ober); Childs, U.S. Patent No. 5,623,545 (Childs); Bruce Schneier, Applied Cryptography (Schneier); Turner et al., U.S. Patent No. 4,896,296 (Turner); and Batcher, U.S. Patent No. 4,314,349 (Batcher). Claim 8 stands rejected under 35 U.S.C. 103 (a) as being obvious in view of Ober, Childs, Schneier, Turner, Batcher and Niehaus et al., U.S. Patent No. 4,399,517 (Niehaus). Claim 16 stands rejected under 35 U.S.C. 103 as being obvious in view of Ober, Childs, Schneier, Turner, Batcher, and Masaki, U.S. Patent No. 4,739,195 (Masaki).

The rejection of claims 1-8 and 14-18 is being appealed.

STATUS OF AMENDMENTS

An amendment received by the U.S.P.T.O. on June 20, 2005, which was filed to correct a previously un-entered amendment, amended claims 1 and 17 to correct minor errors and canceled claims 9-13. This amendment is believed to have been entered because it is only correcting minor errors and canceling claims.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is an apparatus that processes first and second cryptographic has algorithms.

Independent claim 1 recites a register file. The register file has at least five registers for storing chaining variables. In one example set forth in the specification of

the application the register file 12 of FIG. 1 has five registers A, B, C, D, and E. At page 3, lines 11–14, these registers are characterized as being "altered on every clock cycle to contain chaining variables."

Claim 1 further recites a function circuit. The function circuit receives first, second, and third chaining variables and has an output that provides a logical data value. FIG. 1 shows a function circuit 22. At page 3, lines 19-21, describes that function circuit 22 receives data words B, C, and D and that there is a logical data value at the output of function circuit 22. Data words B, C, and D are described at page 3, lines 15-18 as the data words stored in registers B, C, and D, respectively. The data words stored in these registers are described as chaining variables at page 3, line 14.

Claim 1 further recites a first multiplexer. The first multiplexer has an input coupled to the register file for receiving a fourth chaining variable and an output that provides the fourth chaining variable. The output of the first multiplexer provides the fourth chaining variable when the first cryptographic hash algorithm is being processed by the apparatus and a zero value when the second cryptographic has algorithm is being processed by the apparatus. Multiplexer 24 is shown in FIG. 1 as having an input for receiving data word E and an output. Data word E is described at page 3, lines 17 and 18, as the data word stored in register E, which in turn is described as storing a chaining variable at page 3, line 14. At page 7, lines 22-25, the apparatus MDHA 10 of FIG. 1 is described as having a mode in which the MD5 algorithm is selected. When the MD5 algorithm is selected, E is not used and multiplexer 24 provides zero as its output. The MD5 algorithm is described at page 2, lines 23-28, as the Message Digest (MD) 5 cryptographic hash algorithm. In this regard then the MD 5 qualifies as the claimed second cryptographic hash algorithm, which when processed by the apparatus, results in multiplexer 24 providing a zero output. The SHA-1 mode results in data word E being coupled to adder 30 through multiplexer 24 as described at page 5, lines 4-6. As described at page 2, lines 23 and 24, SHA-1 refers to the Secure Hashing Algorithm 1. Thus SHA-1 algorithm qualifies as the claimed first cryptographic has algorithm.

Claim 1 further recites a summing circuit. The summing circuit has a first input coupled to the output of the function circuit and a second input coupled to the output of the first multiplexer, and output coupled to the register file. The first input receives the logical data value. As shown in FIG. 1, adder circuit 30 has one input connected to the output of function circuit 22, has another input connected to multiplexer 24, and has an output coupled to register file 12 through multiplexers 42 and 28, for example. As described at page 5, lines 15-18, the output of adder 30 is provided to register A, which is

part of register file 12, through multiplexers 42 and 28 in the SHA-1 mode. At page 3, lines 19-21, the output of function circuit 22 is described as having the logical data value.

Independent claim 8 is a circuit that is for generating hash values in a first hash mode and a second hash mode. Hash modes include MD5 mode and the SHA-1 mode as described at page 2, lines 26-28. Hash values are described as being generated at page 7, lines 19-21.

Claim 8 recites a storage circuit. As shown in FIG. 1 and described at page 4, line 7, register files 34 and 36 are for storing constant values.

Claim 8 further recites a register array. The register array has registers for storing a message and an output for providing a round dependent data value. Register array 32 is shown in FIG. 1 and in more detail in FIG. 2. FIG. 2 shows that register array 32 provides an output Wt from a multiplexer 124 that is part of register array 32. The output of register array 32 is described at page 9, lines 27 and 28, as being a value that is round dependent. That register array 32 stores a message is described at page 5, lines 26 and 27.

Claim 8 further recites a register file for storing first, second, third, fourth, and fifth chaining variables. Register file 12 of FIG. 1 has five registers A, B, C, D, and E. At page 3, lines 11–14, these registers are characterized as being "altered on every clock cycle to contain chaining variables."

Claim 8 further recites an adder. The adder has a first input for receiving a first set of constant values stored in the storage circuit for the first hash mode and a second set of constant values for the second has mode. The adder has a second input that is coupled to the output of the register array. The adder has a third input for receiving the fifth chaining variable in the second hash mode and a shifted fifth chaining variable in the first hash mode. The adder has a fourth input for receiving a logical function in accordance with the first, second, and third chaining variables. The adder has a fifth input for receiving the fourth chaining variable in the second hash mode and a zero value in the first hash mode. Adder 30, as shown in FIG. 1, has a first input connected to the output of multiplexer 38, which provides a set of constants from the storage circuit made up of registers 34 and 36, which store constant values as described at page 4, line 7. As described at page 2, lines 23 and 28, SHA-1 can be considered a first mode and MD5 can be considered a second mode. Page 4, lines 17-21 describes that the MD5 mode uses register file 36 and the SHA-1 mode uses register file 34. The second input of adder 30 is connected to register array 32 as shown in FIG. 1. The third input of adder 30 is shown in FIG. 1 as being connected to a multiplexer 20 that selects between data word A and shifted data word A. Data word A can be considered the fifth chaining variable because

chaining variables B, C, D, and E are the first, second, third, and fourth chaining variables, respectively. Adder 30 has a fourth input connected to function circuit 22 as shown in FIG. 1. At page 3, lines 19-21, the specification describes that function circuit 22 receives data words B, C, and D, which are chaining variables, and that there is a logical data value at the output of function circuit 22. The fifth input of adder 30 receives chaining variable E in the SHA-1 mode and receives zero in the MD5 mode as described at page 5, lines 4-6.

Independent claim 14 is an apparatus that provides a value of a variable length message in accordance with a first algorithm and a second algorithm.

Independent claim 14 recites a register file. The register file has five registers preset to a first group of values for the first algorithm and to a second group of values for the second algorithm. Register file 12 of FIG. 1 has five registers A, B, C, D, and E. At page 3, lines 11–14, these registers are characterized as being "altered on every clock cycle to contain chaining variables." At page 7, lines 13 and 14, the specification states, "The data words A, B, C, D and E stored in register file 12 are preset to specific values in accordance with the selected algorithm."

Claim 14 further recites a function circuit. The function circuit receives first, second, and third chaining variables and generates a first logical data value for the first algorithm and a second logical data value for the second algorithm. At page 3, lines 19-21, describes that function circuit 22 receives data words B, C, and D and that there is a logical data value at the output of function circuit 22. Data words B, C, and D are described at page 3, lines 15-18 as the data words stored in registers B, C, and D, respectively. Because the data words received by function circuit 22 vary based on the algorithm, the logical data values generated by function circuit 22 will vary as well based on the algorithm.

Claim 14 further recites a storage element. The storage element supplies a first set of constant values for the first algorithm and a second set of constant values for the second algorithm. A storage circuit is made up of registers 34 and 36, which store constant values as described at page 4, line 7. As described at page 2, lines 23 and 28, SHA-1 can be considered a first mode and MD5 can be considered a second mode. Page 4, lines 17-21 describes that the MD5 mode uses register file 36 and the SHA-1 mode uses register file 34.

Claim 14 further recites a summing circuit. The summing circuit has a first input coupled to the output of the function circuit and a second input coupled to the storage element for receiving one of the first and second sets of constant values. Adder 30, as shown in FIG. 1, has an input coupled directly to function circuit 22. Adder 30 has

another input coupled directly to the output of multiplexer 38. Multiplexer 38 provides a set of constants from one or the other of registers 34 and 36, which store constant values as described at page 4, line 7. Page 4, lines 17-21, describes that the MD5 mode uses register file 36 and the SHA-1 mode uses register file 34.

GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

1) Are claims 1-7, 14, 15, and 18 obvious under 35 U.S.C. 103(a) in view of Ober et al., U.S. Patent No. 6,708,273 (Ober); Childs, U.S. Patent No. 5,623,545 (Childs); Bruce Schneier, Applied Cryptography (Schneier); Turner et al., U.S. Patent No. 4,896,296 (Turner); and Batcher, U.S. Patent No. 4,314,349 (Batcher)? Is claim 8 obvious under 35 U.S.C. 103 (a) in view of Ober, Childs, Schneier, Turner, Batcher and Niehaus et al., U.S. Patent No. 4,399,517 (Niehaus)?

2) Is claim 16 obvious under 35 U.S.C. 103 in view of Ober, Childs, Schneier, Turner, Batcher, and Masaki, U.S. Patent No. 4,739,195 (Masaki)?

3) Is dependent claim 17 obvious under 35 U.S.C. 103 in view of Ober, Childs, Schneier, Turner, and Batcher.

ARGUMENT

Arguments for Ground 1

Independent Claim 1

Independent claim 1 stands rejected under 35 U.S.C. 103(a) as being obvious in view of Ober, Childs, Schneier, Turner, and Batcher.

The Examiner's rejection for obviousness used Ober for the general proposition that it was known to have both the SHA-1 hash function and the MD5 hash function available in the same system. The Examiner used Childs as a known way of doing the SHA-1 hash function and Schneier as a known way of doing the MD5 hash function. The Examiner then argued that it would be obvious to realize that there were elements in these two hash functions that were the same. The Examiner further argued that it would be obvious to come up with a single circuit that used the elements that were the same for

both hash functions. The other circuits that were not in common would then be connected as needed using multiplexers, citing Turner and Batcher for this.

In effect the Examiner is saying two things: (1) one of ordinary skill in the art would recognize the benefit of applicants' invention and (2) anytime known elements are used to achieve a benefit that is recognizable to one of ordinary skill in the art, it is obvious for one of ordinary skill in the art to have done so. Applicants agree with the first but not the second. The elimination of the requirement for elements H0-H4 is an obvious reason that applicants' invention is clearly an improvement. This also results in a factor of 5 improvement, 400 cycles compared to 80 cycles, in the number of cycles required to perform the desired hash function. Applicants submit that it is far from obvious to combine FIG. 5 of Childs with FIG. 18.6 of Schneier and obtain FIG. 1 of the application. Although Ober teaches that both hash functions can exist on the same integrated circuit, applicants have not been able to find any suggestion that these two functions can share the same circuitry much less teach which elements can be shared. Accordingly, applicants submit that claim 1 patentably distinguishes over the five reference combination applied by the Examiner.

Dependent claims 2-7

Dependent claims 2 and 3 in particular add elements that point out the implementation that allows for combining the two hash functions in one circuit and the significant improvement over Childs in performance. Of particular significance is the addition of multiplexer 26 in this regard that is very significant in enabling the significant improvement over Childs.

Independent claim 8

The issue is substantially the same as for claim 1.

Independent claim 14 and dependent claims 15 and 18

The issue is substantially the same as for claim 1.

Arguments for Ground 2

Claim 16, dependent on claim 14

Claim 16, among other elements, recites an exclusive OR gate having four inputs. This is supported by XOR 116 of FIG. 2. The Examiner argued that based on Masaki, it would be obvious to substitute the four input exclusive OR gate of Masaki for the exclusive OR gate 605 of FIG. 6 of Childs. Applicants cannot agree at least because Childs is using feedback from PXOR 606 of FIG. 6 to achieve the desired functionality. The substitution suggested by the Examiner is not obvious because it would require a different approach than used by Childs. There's nothing to suggest that the Masaki approach could be used in place of the feedback approach used by Childs.

Arguments for Ground 3

Claim 17, dependent on claim 15 which depends on claim 14

Dependent claim 17 makes it clear that a register array, which is supported by register array 32, is used in common for both hash functions. This element is particularly questionable that it would be obvious to consider it as something that could be used in common. This is particularly true for Schneier. The Examiner simply cited a page 437, line 16 to page 440, line 17 of Schneier. Applicants have not found any reference to a register array in the cited pages of Schneier. Applicants submit that the conclusion that these pages refer to the same circuitry as referenced by elements 602 and 603 in FIG. 6 is far from obvious to one of ordinary skill in the art.

Conclusion

For at least the reasons set forth above, Applicants respectfully submit that the claims of the present application are allowable over the art cited during prosecution.

Respectfully submitted,

James L. Clingan, Jr.

James L. Clingan, Jr.
Attorney for Appellants
Reg. No. 30,163
Ph: (512) 996-6821

Claims Appendix

1. An apparatus for selectively processing first and second cryptographic hash algorithms, comprising:

a register file (12) having at least five registers for storing chaining variables;

a function circuit (22) receiving first (B), second (C) and third (D) chaining variables and an output that provides a logical data value;

a first multiplexer (24) having an input coupled to the register file for receiving a fourth (E) chaining variable and an output that provides the fourth chaining variable when the first cryptographic hash algorithm is being processed by the apparatus and a zero value when the second cryptographic hash algorithm is being processed by the apparatus; and

a summing circuit (30) having a first input coupled to the output of the function circuit for receiving the logical data value, a second input coupled to the output of the first multiplexer, and an output coupled to the register file.

2. The apparatus of claim 1, further comprising:

a barrel shifter (40) having an input coupled to the output of the summing circuit;

an adder (41) having an input coupled to an output of the barrel shifter; and

a second multiplexer (42) having a first input coupled to the output of the summing circuit and a second input coupled to an output of the adder.

3. The apparatus of claim 2, further comprising:

a third multiplexer (26) having a first input coupled to the output of the second multiplexer (42) and a second input coupled to the register file (12) for receiving a fifth (A) chaining variable; and

a fourth multiplexer (28) having a first input coupled to the output of the second multiplexer and a second input coupled to the register file (12) for receiving the third (D) chaining variable.

4. The apparatus of claim 3, wherein the second multiplexer and the fourth multiplexer receive a signal that transfers a summed value from the output of the summing

circuit to the register file when the message digest hardware accelerator is processing an SHA-1 hash algorithm.

5. The apparatus of claim 3, wherein the second multiplexer and the third multiplexer receive a signal that transfers a summed value from the output of the barrel shifter to the register file when the message digest hardware accelerator is processing an MD5 hash algorithm.

6. The apparatus of claim 3, further comprising:

a first shift circuit (16) having an input coupled to the register file for receiving the first (B) chaining variable; and

a fifth multiplexer (14) having a first input coupled to an output of the first shift circuit, a second input coupled to the input of the first shift circuit and an output coupled to the register file for providing the second chaining variable.

7. The apparatus of claim 6, further comprising:

a second shift circuit (18) having an input coupled to the register file for receiving the fifth (A) chaining variable; and

a sixth multiplexer (20) having a first input coupled to an output of the second shift circuit, a second input coupled to the input of the second shift circuit and an output coupled to another input of the summing circuit.

8. A circuit for generating hash values in a first hash mode and a second hash mode, comprising:

a storage circuit (34, 36);

a register array (32) having registers for storing a message and an output for providing a round dependent data value (Wt);

a register file (12) for storing first (B), second (C), third (D), fourth (E) and fifth (A) chaining variables; and

an adder (30) having a first input coupled for receiving a first set of constant values stored in the storage circuit for the first hash mode and a second set of constant values for

the second hash mode, a second input coupled to the output of the register array, a third input coupled for receiving the fifth (A) chaining variable in the second hash mode and a shifted fifth chaining variable in the first hash mode, a fourth input coupled for receiving a logical function in accordance with the first, second and third chaining variables, and a fifth input coupled for receiving the fourth chaining variable in the second hash mode and a zero value in the first hash mode.

14. An apparatus integrated to provide a hash value of a variable length message in accordance with a first algorithm and a second algorithm, comprising:

a register file (12) having five registers preset to a first group of values for the first algorithm and to a second group of values for the second algorithm, the register file storing first (B), second (C), third (D), fourth (E) and fifth (A) chaining variables;

a function circuit (22) receiving first, second and third chaining variables and generating a first logical data value for the first algorithm and a second logical data value for the second algorithm;

a storage element (34, 36) for supplying a first set of constant values for the first algorithm and a second set of constant values for the second algorithm; and

a summing circuit (30) having a first input coupled to the output of the function circuit (22) and a second input coupled to the storage element for receiving one of the first and second sets of constant values.

15. The apparatus of claim 14, further including a register array (32) having a decoder circuit (120) and a plurality of registers for selecting a data word stored in one of the plurality of registers and supplying the data word to an output of the register array when computing the first algorithm.

16. The apparatus of claim 15, wherein the register array further includes:

an exclusive-OR (116) coupled for simultaneously receiving first, second, third and fourth data words stored in the plurality of registers; and

a rotate block (118) having an input coupled to an output of the exclusive-OR and supplying a one bit left circular shift of the data generated by the exclusive-OR to one of the registers in the plurality of registers.

17. The apparatus of claim 15, wherein an output of the register array is supplied from a word wise circular queue when computing the second algorithm.

18. The apparatus of claim 14, wherein the first algorithm is an MD5 algorithm and the second algorithm is an SHA-1 algorithm.

Evidence Appendix

No evidence is submitted in this appendix

Related proceedings Appendix

There are no decisions under this appendix.